B MS1, MS2, and SQT—Three Unified, Compact, and Easily Parsed File Formats for the Storage of Shotgun Proteomic Spectra and Identifications

This chapter describes adaptation of the file infrastructure used by `Sequest` (Eng et al. 1994) to the significant number of spectra produced in a MudPIT experiment.[1] `Sequest` was modified to use und produce the described file formats as described in Sadygov et al. (2002). J. G.'s contribution to the presented material was the `Perl` script `Unitemare` for the conversion of the original `Sequest` file formats into the ones described here and `Perl` scripting for the data presentation by `show`. This chapter was published as

> McDonald, W. H., Tabb, D. L., Sadygov, R. G., MacCoss, M. J. and Venable, J. et al. (2004). MS1, MS2, and SQT—three unified, compact, and easily parsed file formats for the storage of shotgun proteomic spectra and identifications. *Rapid Commun Mass Spectrom*, 18(18):2162–2168.

B.1 Abstract

As the speed with which proteomic labs generate data increases along with the scale

[1] E. g., 6×6000 spectra for a typical six 2 h chromatography cycles MudPIT experiment on ThermoElectron's DecaXP ion trap mass spectrometer, acquiring a maximum of three fragmentation spectra after each full scan.

of projects they are undertaking, the resulting data storage and data processing problems will continue to challenge computational resources. This is especially true for shotgun proteomic techniques that can generate tens of thousands of spectra per instrument each day. One design factor leading to many of these problems is caused by storing spectra and the database identifications for a given spectrum as individual files. While these problems can be addressed by storing all of the spectra and search results in large relational databases, the infrastructure to implement such a strategy can be beyond the means of academic labs. We report here a series of unified text file formats for storing spectral data (`MS1` and `MS2`) and search results (`SQT`) that are compact, easily parsed by both machine and humans, and yet flexible enough to be coupled with new algorithms and data–mining strategies.

B.2 Introduction

Proteomic technologies are helping to change the scale at which biological experiments can be performed. Unfortunately, they also generate such voluminous data that they can result in a computational quagmire. Shotgun proteomic strategies, in which tandem mass spectra (MS/MS) are collected on mixtures of thousands of peptides, require the collection of tens of thousands of spectra (McCormack et al. 1997). Incorporating multidimensional separation strategies such as MudPIT (Multidimensional Protein Identification Technology) can easily balloon this into hundreds of thousands of spectra (Link et al. 1999; Washburn et al. 2001; Florens et al. 2002; and Peng et al. 2003). Spectra must be stored for identification via database search software such as `SEQUEST`. In many cases, multiple identification strategies and even

algorithms are applied in the course of a complete analysis (MacCoss et al. 2002; and Gatlin et al. 2000).

Since its inception as the file format recognized by the `SEQUEST` database search algorithm (Eng et al. 1994), the `DTA` file has seen widespread use as a format to store individual MS/MS spectra. It and the `SEQUEST`–generated `OUT` file were sufficient for experiments producing merely hundreds of spectra. However, as the scope and complexity of these experiments have expanded, the limitations of these files have become increasingly evident; the number of files produced for individual experiments makes directory management problematic, and the storage space wasted in these formats is problematic as well. To deal with some of these limitations we have developed and implemented a new set of unified file formats that are simple, compact, and yet retain their flexibility.

There were several major goals associated with moving towards unified formats. First, unified formats dramatically reduced the number of files required to represent a proteomic data set. This was important because the huge number of files (hundreds of thousands) added to the file servers each week were taxing their stability and exceeded file system limitations. Second, the new formats reduced the amount of storage space used. Many of the individual files were small enough to be below the minimum block size limit for a file (typically 4 or 8). Thus, simply concatenating the files together reduced the total amount of disk space required to store this information. Third, switching to unified file formats enabled greater efficiency in data storage by removing fields that had been repeated in each file and by grouping data that had been distributed to multiple files. Finally, the unified formats were formatted for automated parsing and designed for extensibility. In keeping with this final goal, the formats had to be adaptable to existing programs and able to

accommodate future code developments.

In order to accomplish these goals, we adopted three unified file formats—`MS1`, `MS2`, and `SQT`. All of these store their particular information type for an entire experimental step, e. g., an entire LC/MS/MS experiment or a single salt step from a MudPIT run. The naming convention is simple; the base filename of the instrument–generated file is used with a new extension. For instance, the ThermoFinnigan LCQ file `salt_step.RAW` would generate `salt_step.ms1` and `salt_step.ms2` files and, after database searching, would yield a `salt_step.sqt` file.

The `MS1` file contains full–scan data and is used for analyses that require this type of data such as quantitation or measurement of chromatographic efficiency. The `MS2` file stores MS/MS data and replaces a folder of thousands of `DTA` files; it contains all the spectral information necessary for database searching algorithms. Finally, the `SQT` file unifies the database search results. While initially designed to replace the `SEQUEST OUT` file, it has proven flexible enough to work quite well with other algorithms used in the lab, e. g., `PEP_PROBE` (Sadygov and Yates 2003) and `GutenTag` (Tabb et al. 2003).

B.3 Format Descriptions

These formats were intended to store all necessary information in as compact and accessible a format as possible while retaining human legibility. In general, they contain information generic to all records in a header at the start of the file while data for specific spectra are stored individually through the body of the file. Unless specifically noted, all fields are tab delimited within an individual line. A compre-

hensive description of these formats with examples is available on our website (http://fields.scripps.edu/sequest/unified).

The MS1 format is the simplest of the three. It contains four types of lines: `H`, `S`, `I`, and `[m/z intensity]`. The header is defined by a series of `H` lines. Each line includes a field label and its corresponding value (string, integer, or floating point). The following fields are required in the header: `CreationDate`, `Extractor`, `ExtractorVersion`, and `ExtractorOptions`. The values for these are the date the file was created, the program used, its version, and any specific options used in the program (fig. B.1). Some optional field labels include: `InstrumentType` (ion trap, q–tof, tof–tof, etc.), `InstrumentSN` (serial number), and `Comment` (other information general to the file).

In the body of the file, each full scan in the experiment begins with an `S` line which contains the scan number. This can be followed by the optional `I` line which can be used to store any ancillary information such as retention time (`I RTime 33.2`). Next comes a series of `[m/z intensity pairs]` (space separated) representing the spectral data for that entire full mass scan. This pattern (`S`, `[m/z intensity]`^{n} or `S`, `I`, `[m/z intensity]`^{n}) is repeated for each full mass scan in the experiment. If necessary, multiple `I` lines can be used for a given spectrum (e. g., LC retention time).

The format for the `MS2` file is similar to the `MS1` file except that MS/MS spectra are stored. It shares the `H`, `S`, `I`, and `[m/z intensity]` lines with the `MS1` file but adds two additional lines, `Z` and `D`, to store charge–state–dependent information. The header itself has additional field labels such as `IAnalyzer`, to denote a program which does not consider the charge state of the precursor ion (e. g., spectral quality filtering), and `DAnalyzer`, to denote a program that analyzes charge–state–specific

A

symbol	meaning	generic form			required
H	header	H	[field label]	[corresponding value]	yes
S	scan	S	[first scan]	[second scan]	yes
I	charge independent analysis	I	[field label]	[corresponding value]	
[data]	mass intensity pairs	[m/z]	[intesity]		yes
"H" line required field labels					
field label	**description**				
CreationDate	The date and time on which this file was created.				
Extractor	The name of the software used to create the MS2 file.				
ExtractorVersion	The version number of the Extractor software.				
ExtractorOptions	The options used in running the Extractor software.				
"H" line optional field labels					
field label	**description**				
IAnalyzer	If software to conduct charge state-independent analysis of the spectra is used to analyze the MS2 file				
	(such as removing sparse spectra), the name of the program should be recorded in the header.				
IAnalyzerVersion	The version number of the IAnalyzer				
IAnalyzerOptions	The options used for the IAnalyzer				
InstrumentType	The type of mass analyzer used.				
Comment	Remarks, ownership, and copyright information may be included. Multiple comment lines may be included.				
InstrumentSN	The serial number of the mass spectrometer used.				

B

```
H    CreationDate    07/10/2003
H    Extractor       RAWXtract
H    ExtractorVersion        1.0
H    ExtractorOptions
S    0001    0001
I    RTime   0.01
400.3 828908
400.9 695425
401.9 683908
403.1 556803
405.8 906291
407.2 293691
408.2 2475720
409.4 305360
412.4 324123
413.6 1974477
414.8 2339087
415.6 242656
416.5 1402914
417.4 1635225
418.1 354
419.8 681890
420.5 63
[data continued ...]
```

Figure B.1 `MS1` File Format Description. (A) General description of required fields and format used in the `MS1` file. Both required and optional lines and field descriptions are noted along with a generic pattern for data storage. (B) Example `MS1` file and a partial spectrum. Following the `H` lines of the header each full–scan spectrum begins with an `S` line denoting its scan number. Next, optional `I` lines give additional information about that scan such as retention times. Finally, the spectral data are stored in as series of m/z intensity pairs. This pattern of `S(I)[m/z intensity]`^{n} continues until each full–scan spectrum has been represented.

features (e. g., charge–state discrimination or neutral losses off the precursor). Any specific features noted by the DAnalyzer programs are annotated in the D line following their specific Z line (see below). One advantage of the MS2 file format is that the file format logs which algorithms have been applied serially to the file.

The MS2 file body is structured similarly to the MS1 except for the addition of the Z and D lines. The description of each spectrum begins with the S line which has fields for the [start scan], [end scan], and [precursor m/z]. This can be followed by the optional I line which contains a datum or analytical result that is independent of the charge–state prediction of the precursor, such as a spectral quality score or instructions to the search program not to query this particular spectrum. Next comes the Z line with the [charge state] and [predicted [M+H]+] fields. The optional D line may follow and can be used to store information specific to the charge state of the preceding Z line. This can include annotations of a particular structural feature that might necessitate the use of a different search algorithm, e. g., neutral loss of phosphoric acid off of the precursor as an indication of a phosphorylated peptide. There can be multiple Z and optional D lines for a given spectrum depending on how well the precursor charge state is able to be discriminated. The spectral data are stored in the same manner as the MS1 file with Z and [m/z intensity] stored for every peak in the experimental spectrum. The minimum pattern to represent a spectrum is S, Z, [m/z intensity]^{n}, but, as previously mentioned, can also contain multiple Z lines and the optional I and D lines to encompass additional information. One obvious advantage of this format over the DTA format is that to have the search algorithm to consider an additional charge state, all one has to do is add another Z line rather than producing a separate DTA file (currently each charge state to be considered has a corresponding DTA file), with all of the [m/z

intensity]^{n} information repeated.

The SQT file format is a greater departure from the SEQUEST OUT files it replaces. The design aims were to provide the same information as reported in the OUT file but to do so in a more compact, more easily parsed format while retaining a degree of human legibility. It is comprised of H, S, M, and L lines. The header lines are similar in format to both the MS1 and MS2 files, except that it has its own distinct set of field labels and values. Figure B.2 shows the required fields and an example of how they are employed. Invariant information usually stored in each OUT file is stored just once in the header of the SQT file. The data characterizing a particular identification are stored in a block of lines lower in the file.

Each search result for a spectral entry is denoted by the following generic line pattern: S(M(L)^{k})^{n} (fig. B.3). The S line contains information specific for that spectrum and search. It is followed by an M line which describes a particular matching sequence along with its characteristic scores. Next comes at least one L line that notes which protein in the database contains this particular peptide sequence; there can be multiple L lines depending on how many proteins within the database contain the matched peptide sequence. The M line for the second highest scoring peptide match is followed by its respective L line(s). This pattern of M and L lines continues for as many search results as were set to be stored in the search parameters (typically 5–10). The SQT file also allows the inclusion of a column which stores manual evaluation information. The state can be either the default of U (unevaluated), Y (yes), N (no), or M (maybe). The inclusion of this field allows the SQT file format to store manual validation information that cannot be stored in OUT files.

To institute such file format changes we made modifications to several pre–search (extraction and filtering), searching (SEQUEST), organization and summary

A

symbol	meaning	generic form	required
H	header	`H [field label] [corresponding value]`	yes
S	scan	`S [first scan] [second scan]`	yes
I	charge-independent analysis	`I [field label] [corresponding value]`	
Z	charge	`Z [charge]`	yes
D	charge-dependent analysis	`D [field label] [corresponding value]`	
[data]	mass intensity pairs	`[m/z] [intesity]`	yes

"H" line required field labels

field label	description
CreationDate	The date and time on which this file was created.
Extractor	The name of the software used to create the MS2 file.
ExtractorVersion	The version number of the Extractor software.
ExtractorOptions	The options used in running the Extractor software.

"H" line optional field labels

field label	description
IAnalyzer	If software to conduct charge state-independent analysis of the spectra is used to analyze the MS2 file (such as removing sparse spectra), the name of the program should be recorded in the header.
IAnalyzerVersion	The version number of the IAnalyzer
IAnalyzerOptions	The options used for the IAnalyzer
DAnalyzer	If software to conduct charge state-dependent analysis of the spectra is used to analyze the MS2 file (such as removing possible precursor charge states), the name of the program should be recorded in the header.
DAnalyzerVersion	The version number of the DAnalyzer
DAnalyzerOptions	The options used for the DAnalyzer
SortedBy	If a program is used to sort the spectra, which field are they sorted by?
InstrumentType	The type of mass analyzer used.
Comment	Remarks, ownership, and copyright information may be included. Multiple comment lines may be included.
InstrumentSN	The serial number of the mass spectrometer used.

B

```
H       CreationDate    unknown
H       Extractor       RAWXtract
H       ExtractorVersion        1.0
H       ExtractorOptions
S    0002    0002    534.04
I    RTime   0.02
Z    2       1067.08
Z    3       1600.12
151.0 22869
153.0 14453
155.1 35721
157.1 17102
158.1 12948
166.3 24733
167.0 103456
168.0 11485
171.0 92158
172.1 62988
173.1 136408
[data continued ...]
```

Figure B.2 `MS2` File Format Description. (A) General description of required fields and format used in the `MS2` file. The format follows the general conventions of the `MS1` file format except that MS/MS information is stored and with the addition of the required `Z` lines and the optional `D` lines. (B) Example `MS2` file and partial spectrum. As with the `MS1` file, each spectral description in the `MS2` file begins with an `S` line. `Z` lines denote which charge states are to be considered for the spectrum. Like the `MS1` file, the spectrum itself is represented as a series of `[m/z intensity]` pairs. Optional `I` and `D` lines can be used to store charge–state–independent and charge–state–dependent information, respectively. The general pattern `S(I)[Z(D)]`^{k}`[m/z intensity]`^{n} continues until all MS/MS spectra are represented.

A

symbol	meaning	generic form	required
H	header	H [field label] [corresponding value(s)]	yes
S	spectrum	S [low-scan] [high-scan] [charge] [process time] …	yes
		… [server] [obs. Mass] [lowest SP] [# sequences match]	
M	match	I [rank by Xcorr] [Rank by Sp] [calculated mass] …	yes
		… [DeltCN] [Xcorr] [Sp] [matched ions] [expected ions] …	
		… [sequence matched] [validation status]	
L	locus	L [locus] [description (if contained in database)]	yes

"H" line required field labels	
field label	**description**
SQTGenerator	Which program created this SQT file?
SQTGeneratorVersion	What revision of the program was used to produce this SQT?
Database	What was the path and filename of the database used to produce this SQT? Multiple databases may be listed here.
FragmentMasses	Were average or monoisotopic residue masses used to predict the fragment ion masses?
PrecursorMasses	Where average or monoisotopic residue masses used to predict the precursor ion mass? ("AVG" and "MONO")
StartTime	When was this file initiated?
StaticMod	If any nonstandard amino acid masses were used in identification, they must be listed here.If multiple static
	modifications were used, multiple StaticMod lines should be present.
DynamicMod	If any dynamic modifications were sought in identification, they must be listed here.

"H" line optional field labels	
field label	**description**
Comment	Remarks, ownership, and copyright information may be included. Multiple comment lines may be included.
DBSeqLength	How many amino acids appear in the sequence database?
DBLocusCount	How many protein sequences appear in the sequence database? This field must follow the same rules as DBSeqLength.
DBMD5Sum	To ensure that the same database is currently present as when this search was conducted, the MD5Sum of the
	database may be stored in the SQT header. This field must follow the same rules as DBSeqLength.
SortedBy	If the IDs in this file have been sorted, which field was used?
Alg-	Other fields may be added to the SQT header. Any algorithm-specific fieldname should begin with "Alg-" to prevent
	parsing errors. Field names may not included whitespace characters.

B

```
H    SQTGenerator       SEQUEST
H    SQTGeneratorVersion      2.7
H    Comment SEQUEST was written by J Eng and JR Yates, III
H    Comment SEQUEST ref. J. Am. Soc. Mass Spectrom., 1994, v. 4, p. 976
H    Comment SEQUEST ref. Eng,J.K.; McCormack A.L.; Yates J.R.
H    Comment SEQUEST is licensed to Finnigan Corp.
H    Comment Paralellization Program is run_ms2
H    Comment run_ms2 was written by Rovshan Sadygov
H    StartTime    07/10/2003, 08:00 PM
H    EndTime      07/10/2003, 08:00 PM
H    Database     /wfs/dbase/nci/MAPdb.nci
H    DBSeqLength  901292
H    DBLocusCount 1180
H    PrecursorMasses    AVG
H    FragmentMasses     AVG
H    Alg-PreMassTol     3.000
H    Alg-FragMassTol    0.0
H    Alg-XCorrMode      0
H    StaticMod    C=160.139
H    DiffMod      ST*=+80.000
H    Alg-MaxDiffMod  4H      Alg-DisplayTop  5
H    Alg-IonSeries   0 1 1 0.0 1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0^M
S    0012         0012  1     1     shamu40     659.57      3716.5      73.8  111330
M    1       1      658.600   0.0000      0.0000      168.4    7    10    S.LSSNGT*.N     U
L    GP:AF075587_1
M    1       2      656.627   0.0000      0.0000      155.1    7    12    P.AALGSAS*.A    U
L    SW:DHE3_HUMAN
M    1       3      657.656   0.0000      0.0000      147.4    7    10    K.VTGLST*.R     U
L    GP:HS511B24_3
M    1       4      656.673   0.0000      0.0000      146.5    8    16    K.PTGGPGGGG.T   U
L    GP:HUMRCK_1
L    PIR2:S22651
L    SW:DDX6_HUMAN
```

Figure B.3 `SQT` File Format Description. (A) General description of lines, required fields, and generic format for the `SQT` file. (B) An example `SQT` file header with a portion of search results for an MS/MS spectrum. The search results for each spectrum start with an `S` line which contains the scan numbers and certain other metrics relating to that spectrum. The first `M` line gives the search results for the highest scoring peptide in the database and is followed by one or more `L` lines that give locus names for the proteins in the database in which that peptide could be found. `M` and `L` line combinations are given for the remaining recorded search results for that spectrum. Results for each searched spectrum are recorded in the general format: $\texttt{S}[\texttt{ML}^{k}]^{n}$.

(DTASelect, Tabb et al. 2002), and visualization (results display) programs. For spectral extraction we have the Linux–compiled, makems2 to extract MS/MS from a.dat (ICIS) file into an MS2 format and perform limited charge–state selection and filtering. It is basically a unified format version of the extractms program. For spectral extraction directly from RAW files, we have MSMaker which extracts both MS and MS/MS spectra, but does only limited charge–state selection. Charge–state selection is done primarily using 2to3u (Sadygov et al. 2002). DTASelect has been designed to accommodate a variety of file formats, including these new ones. For visualization purposes we developed a new CGI, show (http://fields.scripps.edu /sequest/show/index.html), which gathers information from both the SQT and MS2 files and passes them to an applet version of the DTASelect ion display graphical interface. It is also back compatible with DTA, OUT, and a variety of intermediate file formats. Finally, we developed a PERL script, Unitemare, which transcodes previously searched DTAs and OUTs into the new unified formats. With the exception of SEQUEST, these programs are freely available to academic and other nonprofit groups; see the group website for details (http://fields.scripps.edu).

For performance comparisons a single MudPIT cycle was chosen from a six–cycle analysis of a previously described protein mixture (McDonald et al. 2002). These data were extracted either into DTA or MS2 format without filtering and only rudimentary charge–state discrimination. The 2886 nonblank MS/MS spectra generated a total of 5642 DTA files to be searched (as a result of the need to store a separate DTA file if multiple charge states were to be considered). Both formats were searched using SEQUEST against the same database using identical settings to generate OUT and SQT files. Disk usage measures were performed either using the Linux du -b command or folder properties in WindowsXP.

B.4 Results and Discussion

One of the primary design goals for these files was to reduce the number of files which must be stored on the hard drive. Clearly, this was accomplished since the MS/MS spectra and search results of a typical MudPIT cycle were able to be stored in two files rather than the 5642 `DTAs` and their corresponding 5642 `OUT`s present in the example file. While it is difficult to quantify the impact that this has had on the stability of our file servers, anecdotally we noted a dramatic increase in uptime with a concomitant decrease in errors (primarily network file system, NFS, errors). We experienced this increased stability in spite of going from 1/3 terabytes of data to > 1.5 terabytes of data stored on our two `Linux` file servers.

A potential problem with aggregating results into a single file is that accessing a specific spectrum or search result can be much slower than having them split out as single files. We tried to address this in two ways. One was through the line tags that preclude the need for complex matching strategies across an entire line. For instance to find a particular spectrum, one need only consider lines starting with the S token. Another is that the files were kept as streamlined as possible. Sorting the files to place the highest scoring identifications and their corresponding spectra to the top of the files (`SQTSort`, http://fields.scripps.edu/sequest/SQTSort.html) enabled even faster access to the most relevant data. For speed and even more compactness, we are exploring the possibilities of having indexed binary versions of these file formats. In addition, a conscious decision was made to store only the necessary information and not bloat them with information that, while likely to be useful at some future point, was either present in the initial instrument file or more efficiently stored in a separate file within a given experimental folder.

The next goal was to reduce the total disk space required to store the files. Simply concatenating the spectral and search result files would be predicted to help

Table B.6a Savings Per File Type

		Bytes	Saved
MS/MS Data	DTAs	35 115 008	

Table B.7b Savings Per File Type

	MS2	10 207 232	70.93 %
Search Results	OUTs	23 355 392	
	SQT	3 858 432	83.48 %

since many of these files were smaller than the block size or minimum file size. This could be seen by measuring the total disk usage for a folder of DTA and OUT files, 58 908 672 bytes total, versus the total usage if all the DTA files and OUT files were put together into two files, 35 729 408 bytes. However, the redundancy of the headers in the OUT files and the file repetitions needed to represent multiple charge states allowed a total saving of about 75 k% for the SQT and MS2 file formats (Table B.8). The largest reduction in disk space requirements was seen in going from the OUT files to the SQT file, an 83 % saving (Table B.6). Similar differences were seen when the files were stored on a WindowsXP machine (NTFS formatted partitions) (Table B.8). These savings scaled proportionally to the number of LC/MS/MS runs (data not shown). Since it is now trivial to collect 100s of gigabytes of data in a relatively short period of time, being able to store data in one–quarter of the space is significant, especially for those academic labs that are unable to afford or maintain multiterabyte storage arrays. As newer and faster instruments emerge these considerations will become even more significant; for instance, the next generation ThermoFinnigan linear ion trap (LTQ) collects about four and a half times as many spectra as the LCQ in the same amount of time.

Another advantage we have noted is that programs which have to read through every result in a given experiment, such as DTASelect, are able to parse through

Table B.8a Total Savings

	Linux	Saved	WinXP	Saved
DTAs and OUTs	58 462 208		59 469 923	

Table B.9b Total Savings

Concatenated	35 274 752	39.66 %	36 630 528	38.40 %
MS2 and SQT	14 061 566	75.95 %	14 987 264	74.80 %

these files more quickly. For those same 5642 OUT files, DTASelect required 4.2 s to read through and gather all the necessary information. The corresponding SQT files were parsed in a less than a second. Again, while not substantial for a single file, this parsing can take a great deal of time when the dataset consists of > 50 MudPIT cycles which in turn were searched against multiple databases (Florens et al. 2002). In comparison to their predecessors, the design of the unified file formats makes it relatively easy to develop software to read them, especially when compared to the difficulty of dealing with all of the subtle differences in OUT files produced by different SEQUEST revisions.

These file formats also allowed for a more streamlined, logical, and flexible workflow. First, the MS2 file logs the serial application of multiple programs during the data analysis workflow. After initial extraction of the various files, these could include spectral quality filtering/scoring, charge–state selection, and feature annotation. Which programs have been run is stored as header information and the I and D lines provide the flexibility to annotate specific spectra and/or specific charge states of those spectra. This expansion room without compromising file size is an important feature that was missing from the original DTA file format.

Another workflow advantage can be seen when one wishes to analyze the same data versus multiple databases or for a variety of posttranslational modifications

(e. g., see MacCoss et al. 2002). Instead of having to search copies of the `DTA` files it becomes quite practical to use symbolic links back to the original `MS2` files. Even more dramatic space savings can be realized under such a scenario. Using symbolic links to the `MS2` files, six different searches could be performed on our example LC/MS/MS run with a total savings of 94 % over `DTA`s and `OUT`s (352 megabytes vs. 22 megabytes). In fact, after all of these searches have been performed, it is possible to collate the various answers back into a single aggregate `SQT` file. Extensive analysis of complex datasets requires flexible formats to bring these results together into an easily digested final output. The `MS1` file format, for instance, could be used to extract full–scan chromatograms of individual peptides for purposes of quatitation or characterizing chromatographic efficiency.

Ongoing discussions seek to standardize file formats for proteomic data, with the ultimate goal of moving towards a common database schema that can be employed globally (Orchard et al. 2003; and Taylor et al. 2003). However, there is an evident need for an intermediate step moving from either the single spectrum or proprietary instrument manufacturer formats to this ultimate goal. Several groups are proposing moving towards common `XML` (extensible markup language) formats in order to store all the data concerning a particular experiment. There are many advantages to this idea in terms of tools available to deal with `XML` data, and of course, a common language spoken by all proteomics labs. However, `XML` files typically spend many bytes on formatting information, potentially increasing rather than decreasing storage capacity requirements. The extensibility to new instruments and experimental strategies possible with `XML` formatting may not prove an adequate gain for the cost in file size.

The `MS1`, `MS2`, and `SQT` file formats do not meet all of the goals of these upcoming

database and XML standards, and are not intended to substitute for them. However, since they are compact, flexible, easily parsed, and mature in their implementation, we propose that they can serve a very useful role in the proteomic mass spectrometry community. They should be particularly appealing to small–scale labs that are still able to generate large volumes of data, but have necessarily limited computational and storage resources. The ease with which these files can be parsed and the existing suite of tools under continuing development in our group and others make them an appealing platform. The use of tab–delimited text files makes the creation of translation software to produce other formats of data trivial, allowing export to whatever industry standards are ultimately adopted. We propose that they are a viable alternative to an XML–based single file format because of advantages in disk space required, developmental flexibility, and ease in later translation to an industry–standard XML or database format for dissemination and sharing of data.

B.5 References

Eng, J. K., McCormack, A. L. and Yates III, J. R. (1994). An approach to correlate tandem mass–spectral data of peptides with amino–acid sequences in a protein database. *J Am Soc Mass Spectr*, 5(11):976–989.

Florens, L., Washburn, M. P., Raine, J. D., Anthony, R. M. and Grainger, M. et al. (2002). A proteomic view of the *Plasmodium falciparum* life cycle. *Nature*, 419(6906):520–6.

Gatlin, C. L., Eng, J. K., Cross, S. T., Detter, J. C. and Yates III, J. R. (2000). Automated identification of amino acid sequence variations in proteins by HPLC/microspray tandem mass spectrometry. *Anal Chem*, 72(4):757–63.

Link, A. J., Eng, J., Schieltz, D. M., Carmack, E. and Mize, G. J. et al. (1999). Direct analysis of protein complexes using mass spectrometry. *Nat Biotechnol*, 17(7):676–82.

MacCoss, M. J., McDonald, W. H., Saraf, A., Sadygov, R. and Clark, J. M. et al. (2002). Shotgun identification of protein modifications from protein complexes and lens tissue. *Proc Natl Acad Sci USA*, 99(12):7900–5.

McCormack, A. L., Schieltz, D. M., Goode, B., Yang, S. and Barnes, G. et al. (1997). Direct analysis and identification of proteins in mixtures by LC/MS/MS and database searching at the low–femtomole level. *Anal Chem*, 69(4):767–76.

McDonald, W. H., Ohi, R., Miyamoto, D. T., Mitchison, T. J. and Yates III, J. R. (2002). Comparison of three directly coupled hplc ms/ms strategies for identification of proteins from complex mixtures: Single–dimension lc–ms/ms, 2–phase mudpit, and 3–phase mudpit. *Int J Mass Spectrom*, 219(1):245–251.

McDonald, W. H., Tabb, D. L., Sadygov, R. G., MacCoss, M. J. and Venable, J. et al. (2004). MS1, MS2, and SQT—three unified, compact, and easily parsed file formats for the storage of shotgun proteomic spectra and identifications. *Rapid Commun Mass Spectrom*, 18(18):2162–2168.

Orchard, S., Hermjakob, H. and Apweiler, R. (2003). The proteomics standards initiative. *Proteomics*, 3(7):1374–6.

Peng, J., Schwartz, D., Elias, J. E., Thoreen, C. C. and Cheng, D. et al. (2003). A proteomics approach to understanding protein ubiquitination. *Nat Biotechnol*, 21(8):921–6.

Sadygov, R. G., Eng, J., Durr, E., Saraf, A. and McDonald, H. et al. (2002). Code developments to improve the efficiency of automated MS/MS spectra interpretation. *J Proteome Res*, 1(3):211–5.

Sadygov, R. G. and Yates III, J. R. (2003). A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Anal Chem*, 75(15):3792–8.

Tabb, D. L., McDonald, W. H. and Yates III, J. R. (2002). DTASelect and Contrast: Tools for assembling and comparing protein identifications from shotgun proteomics. *J Proteome Res*, 1(1):21–6.

Tabb, D. L., Saraf, A. and Yates III, J. R. (2003). GutenTag: High–throughput sequence tagging via an empirically derived fragmentation model. *Anal Chem*, 75(23):6415–21.

Taylor, C. F., Paton, N. W., Garwood, K. L., Kirby, P. D. and Stead, D. A. et al. (2003). A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nat Biotechnol*, 21(3):247–54.

Washburn, M. P., Wolters, D. and Yates III, J. R. (2001). Large–scale analysis of the yeast proteome by multidimensional protein identification technology. *Nat Biotechnol*, 19(3):242–7.